



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/671,056

09/25/2003

Daniel Alan Brokenshire

AUS920030701US1

5833

40412

7590

09/03/2008

IBM CORPORATION- AUSTIN (JVL)  
C/O VAN LEEUWEN & VAN LEEUWEN  
PO BOX 90609  
AUSTIN, TX 78709-0609

EXAMINER

WEI, ZHENG

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

09/03/2008

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/671,056	<b>Applicant(s)</b> BROKENSIRE ET AL.	
	<b>Examiner</b> ZHENG WEI	<b>Art Unit</b> 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 06 June 2008.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1,3-8,10-14 and 16-20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,3-8, 10-14 and 16-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                       | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>07/12/08, 05/31/08, 04/30/08, 03/30/08</u> .                  | 6) <input type="checkbox"/> Other: _____                          |



## **DETAILED ACTION**

### ***Remarks***

1. This office action is in response to the amendment filed on 06/06/2008.
2. Claims 1, 8 and 14 have been amended.
3. Claims 1, 3-8, 10-14 and 16-20 remain pending and have been examined.

### ***Information Disclosure Statement***

4. The information disclosure statements filed on 07/12/2008, 05/31/2008, 04/30/2008 and 03/30/2008 have been placed in the application file, which the information referred to therein has already been considered.

### ***Response to Arguments***

5. Applicant's arguments, see pages 11-12, filed 06/06/2008 with respect to the rejection(s) of claims under 35 U.S.C. § 103 has been fully considered but are not persuasive. For example:
  - At page 11, last paragraph, the Applicant submits that neither Oram nor Stallman teach or suggest, either alone or in combination with each other, all the limitations included in Applicants' claim 1 as amended. Because Oram's library file is created from two already created object files, and in no way is the same as "compiling the source code, which directly creates an object files" as claimed by Applicants.

Art Unit: 2192

However, the Examiner's position is that both prior art Oram/Stallman and the Applicant's invention direct to generate an object file by compiling source code for a plurality of heterogeneous processor types by using different implementation methods wherein the Applicant's invention directly creates an object file during compilation and the prior art's method uses separate steps to create an object file after compilation. Moreover, Oram also discloses a macro can be used to implement a make file to combine compilation and create an object file (library file) (see for example, p.34, make file example "prog"). Therefore all the steps in Oram/Stallman can be executed by running only one macro instruction and thus is similar to the method of the Applicant's invention.

### ***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1, 3, 6-8, 10, 13-14 and 16, 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Oram (Oram et al., Managing Projects

Art Unit: 2192

with make) in view of Stallman (Richard M. Stallman, Using the GNU Compiler Collection for GCC3.1)

**Claim 1:**

Oram discloses a method for compiling source code, said method comprising:

- receiving source code that includes a plurality of source code subtasks (see for example, p.79, example make file receives source code subtasks trac.c and main.c)
- independently selecting compile option (see for example, p.79, lines 9-11, define the proper compile option symbols in CFLAGS for each source file; also see example make file and related text)

But does not explicitly disclose independently selecting a processor type from the plurality of heterogeneous processor types for each of the plurality of source code subtasks. However, Stallman in the same analogous art of source code compilation discloses compilation option of plurality of heterogeneous processors type (see for example, p.10-16, a list of machine dependent options for different processor types, e.g., p.12 lines 43-46, a set of option can be selected for MIPS processor).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to select different option according to different target process type. One would have been motivated to do so to select the proper symbols in CFLAGS to generate correct type

Art Unit: 2192

executable code that can be run at different processes as suggested by Oram.

Oram further discloses

- selecting a first processor type from the plurality of heterogeneous processor types for a first source code subtask included in the source code (see for example, p.79, example code “make trac.o ‘CFLAGS = -DSTATS -DBSD’; cc -DSTATS -DBSD -c trac.c”. The CFLAGS option can also include -m option as Stallman disclosed above for process type); and
- selecting a second processor type from the plurality of heterogeneous processor types for a second source code subtask included in the source code, wherein the second processor type is different than the first processor type(see for example, p.79, example code “make main.o ‘CFLAGS = -DBSD’; ‘cc -DBSD -c main.c’”. The CFLAGS option can also include -m option as Stallman disclosed above for process type); and

after the processor type sections, compiling the source code, which directly creates an object file (library file) that includes a first object code corresponding to the first source code subtask and second object code corresponding to the second source code subtask, wherein the first object code is adapted to be processed by the first processor type and the second object code is adapted to be processed by the first processor type and the second object code is adapted to be processed by the second

Art Unit: 2192

processor type. (see for example, p.33, example code, “ar r libops interact.o sched.o”. Two object files interact.o and sched.o are combined to generate a libops library file. The library file “libops” is considered as single file physically contains the same content and format of those two object files interact.o and shced.o. Furthermore, the “libops” performs the same way as one object file which contains two objects file or two separate object files during the linking/loading processes; also see p.34, example of make file “prog”). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to amend the Oram's make file macro to specify processor types for different source code subtasks using Stallman and further create a single object file (library file).

**Claim 3:**

Oram and Stallman discloses the method as described in claim 1 above, Oram also discloses wherein the selection of the first processor type is performed during compilation, the method further comprising:

- retrieving the first source code subtask from the plurality of source code subtasks (see for example, p.79, example make file receives source code subtasks trac.c and main.c);
- determining whether the first source code subtask includes a program directive (see for example, p.78, Conditional compilation, through preprocessor directives like #ifdef and #ifndef); and



- performing the selection of the first processor type in response to the determination (see for example, p.79, example code “S make full\_test”).

But does not explicitly disclose determining whether the first source code subtask includes a program directive corresponding to one of the plurality of processors. However, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use `#ifdef` and `#ifndef` directives to define process type option for each source code subtask. One would have been motivated to do so to run on different hardware or operating system as suggested by Oram (see for example, p.78, section “Compiler Option and `#ifdef` directives”, first paragraph, “some of the alternatives reflect the need to compile and run on different hardware or operating systems”)

**Claim 6:**

Oram and Stallman disclose the method as described in claim 1 above;

Oram further discloses the method comprising:

- retrieving the first source code subtask from the plurality of source code subtasks (see for example, p.79, example make file receives source code subtasks `trac.c` and `main.c`);
- identifying one or more operations included in the first source code subtask (see for example, p.79, example code about “make”, after running “\$make full\_test”)

Art Unit: 2192

- matching one or more of the operations with one of the processor types from the plurality of heterogeneous processor types (see for example, p.79, example code for compiling trac.c and main.c file using different option); and
- performing the selection of the first processor type in response to the matching ((see for example, p.79, example code for compiling trac.c and main.c file using different option and generating different trac.o and main.o files).

**Claim 7:**

Oram and Stallman disclose the method as described in claim 1 above, Stallman also discloses the method as described in claim 1 further comprising:

- receiving a processor-specific command, the processor specific command (see for example, p.75, section 3.17 Hardware Models and Configurations, lines 11-14, “In addition, each of these target machine types can have its own special options, starting with ‘-m’ to choose among various hardware models or configurations – for example, 68010 vs 68020...” )

Oram and Stallman further disclose following

- identifying a processor type from the plurality of heterogeneous processor types (see for example, p.79, example code make use the

Art Unit: 2192

process type option which is defined in CFLAS as addressed above);

and

- performing the selection of the first processor type based upon the processor-specific command (see for example, p.79, "\$make full\_test" and related text; also see p.79, "make passes the right -D option to each command" and related text)

**Claims 8, 10 and 13:**

Claims 8, 10, 13 are system version for performing the claimed method as in claims 1, 3 and 6 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly a computer system would need to run and/or practice such function steps disclosed by Stallman and Oram. Thus, they also would have been obvious.

**Claims 14, 16 and 19-20:**

Claims 14, 16 and 19-20 are computer program products version of the claimed method, wherein all claimed limitation functions have been addressed in claims 1, 3, 6-7 above respectively. It is well known in the computer art that such method steps can be implemented as computer program and can be practiced and /or stored on a computer operable media. Thus, they also would have been obvious in view of Stallman and Oram's teachings.

8. Claims 4, 5, 11, 12, 17 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Oram (Oram et al., Managing Projects with make) in view of Stallman (Richard M. Stallman, Using the GNU Compiler Collection for GCC3.1) in further view of Per Bothner (Compiling Java with GCJ)

**Claim 4:**

Oram and Stallman disclose the method as described in claim 1 above, Oram further discloses the method as described in claim 1 further comprising:

- retrieving the first source code subtask from the plurality of source code subtasks (see for example, p.79, example make file receives source code subtasks trac.c and main.c); and
- compiling the first source code subtask use c compiler (cc/gcc), the compiling resulting in object file (see for example, p.79 example code “make trace.o ‘CFLAGS - -DSTATS –DBSD”).

But neither of them discloses compiling resulting in byte code. However, Bothner in the same analogous art of source code compiling, discloses using GCJ compiler to compiling Java code to generating byte code (see for example, p.3, section “compiling a Java Program with GCJ”).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Java compiler instead of C compiler to compile Java source code and generating byte code results.

**Claim 5:**

Oram and Stallman disclose the method as described in claim 4, but does not disclose said method further comprising: sending the byte code to a client over a computer network, wherein the byte code is adapted to be translated into client-specific object code by the client whereby the client-specific object code is formatted based upon a processor type that is located at the client. However, it is well known in the computer art at the time the invention was made that said byte code, as a type of computer program code can be sent and/or retrieved over computer network using any transmission protocols, e.g., TCP/IP. It is also well known in the computer art that byte code can be interpreted and executed at client machine by using client's Just-In-Time compiler to translated into client specific object code. Therefore, claim 5 is unpatentable over Oram, Stallman Bothner and well-known feature discussed above.

**Claims 11 and 12:**

Claims 11 and 12 are system version for performing the claimed method as in claims 4 and 5 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly a computer system would need to run and/or practice such function steps disclosed by Stallman, Oram and Bothner Thus, they also would have been obvious.

**Claims 17 and 18:**

Claims 17 and 18 are computer program products version of the claimed method, wherein all claimed limitation functions have been addressed in claims 4 and 5 above respectively. It is well known in the computer art that such method steps can be implemented as computer program and can be practiced and /or stored on a computer operable media. Thus, they also would have been obvious in view of Stallman, Bothner and Oram's teachings.

***Conclusion***

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Applicant's arguments with respect to claims rejection have been considered but are not persuasive. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period

Art Unit: 2192

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571) 270-1059 and Fax number is (571) 270-2059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tuan Q. Dam/  
Supervisory Patent Examiner, Art Unit 2192